**Assignment 22**

Aim: Create Flutter application to demonstrate REST API.

Code:-

**main.dart**

import 'package:flutter/material.dart';

import 'package:restapidemo/remote\_services.dart';

import 'package:restapidemo/user.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),

useMaterial3: true,

),

home: const MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({super.key, required this.title});

final String? title;

@override

State<MyHomePage> createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

List<User>? users;

var isLoaded = false;

@override

void initState() {

super.initState();

getData();

}

getData() async{

users = await RemortServer().getUsers();

if (users!=null){

setState(() {

isLoaded=true;

});

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Theme.of(context).colorScheme.inversePrimary,

title: Text("Rest API Program"),

),

body: Visibility(

visible: isLoaded,

child: ListView.builder(

itemCount: users?.length,

itemBuilder: (context,index){

return Container(

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(users![index].name,

maxLines: 2,

overflow: TextOverflow.ellipsis,),

],

), //Column

);

}

),

replacement: const Center(child: CircularProgressIndicator()),

)

);

}

}

**user.dart**

import 'dart:convert';

List<User> userFromJson(String str) => List<User>.from(json.decode(str).map((x) => User.fromJson(x)));

String userToJson(List<User> data) => json.encode(List<dynamic>.from(data.map((x) => x.toJson())));

class User {

int rno;

String name;

User({

required this.rno,

required this.name,

});

factory User.fromJson(Map<String, dynamic> json) => User(

rno: json["rno"],

name: json["name"],

);

Map<String, dynamic> toJson() => {

"rno": rno,

"name": name,

};

}

// generated using:- https://app.quicktype.io/

**RemortServer.dart:-**

import 'package:restapidemo/user.dart';

import 'package:http/http.dart' as http;

class RemortServer{

Future<List<User>?> getUsers() async{

var client = http.Client();

var url = Uri.parse("https://mocki.io/v1/448c145c-a19e-4506-a6a2-f0c3aebbd60d");

var response = await client.get(url);

if(response.statusCode==200){

var json = response.body;

return userFromJson(json);

}

}

}

Output:-

